

IBM SanFrancisco: Moving into the marketplace

by V. M. Johnson
D. K. Austreng

Two years ago, the IBM Systems Journal published an issue containing papers on the content and use of the IBM SanFrancisco™ product. Since that time we have seen changes: in the product, in how customers are using the product, and in the marketplace in general. This essay provides a brief review of some of these changes and introduces the papers in this issue related to the SanFrancisco theme.

In May, 1998, when the *IBM Systems Journal* published the previous set of papers on IBM SanFrancisco*, the product had been on the market for only a few months. Customers were just starting to learn to use the product. One paper¹ provided an overview of IBM SanFrancisco and its programming models. Customers faced the task of learning how to understand the models and code that were delivered to them. Our earlier papers addressed enterprise modeling,² evolving from traditional to object technology,³ and how to introduce SanFrancisco into traditional development approaches.⁴ Our customers needed to determine how to take advantage of the SanFrancisco function and to save time in their development processes. A technical note explained how to use SanFrancisco together with VisualAge* for Java** to build applications.⁵ Other papers discussed these questions and speculated on how the product would be used to build applications.^{2,6}

As a result of the early customer experience, work was done to make the product easier to learn and understand. Several user guides were developed to help application developers find what was provided in the frameworks and how to use it. We have learned that it is still difficult for someone with no object-oriented skills to learn to use SanFrancisco. We rec-

ommend that some customers take advantage of consultants who have SanFrancisco skills to help them make the transition.

Work has also been done to improve the SanFrancisco product. Components have been delivered for several additional domains, including combined ledger, order management, and warehouse management. These provided additional functions to application developers. Other work has been done to improve the performance of SanFrancisco-based applications. This resulted from improvements to Java technology as well as product-specific work on SanFrancisco. In the industry, Java virtual machines have been improved and better approaches to client architectures have been discovered. On SanFrancisco, better designs and implementations of the components, and applications based on them, have resulted in performance levels that meet real business demands.

At this time, IBM SanFrancisco customers are starting to deliver applications and install them in real customer locations. We are still in the early stages of this process, but some customers are already supporting their businesses on SanFrancisco-based applications, and others are in the final stages of parallel testing prior to "cutting over." The number of successfully installed applications should increase rapidly.

©Copyright 2000 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

The market continues to change

With the focus on “wiring the enterprise,” software development has shifted its focus from the desktop (or “Webtop”) to the server space, promising software that functions as a bridge to legacy systems. This trend, coupled with the fast-paced pervasive computing trend, delivers on a hope that many believed the personal computer revolution could fill: seamless communication throughout the organization. As

SanFrancisco defines a set of object-oriented interfaces to an infrastructure that can be used to build e-business applications.

application breakthroughs continue to occur at an increasing rate, markets are dislocated and new solutions race to fill the new technological space, each competing to become the standard.

The e-business space is rapidly becoming essential to all businesses. No longer just a nicety, it is rapidly becoming a competitive necessity for supporting key business applications on the Internet. Applications range from direct customer sales via on-line catalogs, to providing better access to information to customers, to use of the network to connect to a server and provide input to an application. The industry has learned that not all application architectures function well in this environment. Currently thin-client approaches are gaining in popularity and provide the level of responsiveness that is required.

One of the contributions of SanFrancisco was the definition of a set of object-oriented interfaces to an infrastructure that could be used to build e-business applications. The industry has welcomed this concept and included knowledge learned from SanFrancisco in the Enterprise JavaBeans** (EJB) standard. EJB is now competing more directly with Microsoft's COM (Common Object Model). Several companies (including IBM) are involved in discussions to enhance the EJB interfaces. It will take time for the final set of interfaces to become available. In the meantime, it seems likely that individual infrastructure vendors will continue to provide functions above and beyond the standard interfaces. This means that

developers may have to make a choice between not having and using key functions, or writing applications now and having to port them to a full standard infrastructure when it is finally available.

Because of the emergence of the standard interfaces, a successful commercial market is now beginning to develop for server-side Java applications. As the market for EJB application servers used in e-business emerges, it will present a significant commercial opportunity to component vendors.

The Extensible Markup Language (XML) is perhaps the fourth component integration technology. While CORBA** (Common Object Request Broker Architecture**), EJB, and COM+ are obvious contenders in this technology race, XML is emerging to provide the essential communication piece needed for components to interoperate over the Internet. Although originally designed to overcome problems in HTML (HyperText Markup Language), XML is rapidly becoming the technology of choice for integrating heterogeneous component-based systems. XML allows the structure of the data to be represented separately from the data values. This representation is system- and platform-independent. It is easy to understand and to process programmatically.

XML allows definition of a minimal “wire representation” for information and message interchange. This is needed to ensure that components can communicate. At the heart of the XML is an extremely simple specification; it provides the syntactic ground rules for forming valid XML messages. While the World Wide Web Consortium (W3C)⁷ is evolving additional standards on top of XML, such as XLink⁸ and XML Schema,⁹ the foundation of XML syntax has remained relatively stable. The base XML syntax is flexible and adaptable to many applications. Several uses are emerging, for example, defining interfaces between applications or components, or presenting data for display in thin-client applications. Perhaps XML will emerge as the domain-neutral solution for interoperation.

Another emerging practice is the building of applications using predefined components. The software industry has long hoped for interoperable, reusable, and scalable components. Perhaps the Java language with its cross-platform capability, the need for faster software assembly, or simply the characterization of software as a reusable asset drives this emerging market. Whatever the cause, this trend will profoundly change the landscape of traditional software and

usher in new ways to create applications. To build an application by assembling predefined components, instead of by building code that performs the functions, it helps to have standard interfaces for common infrastructure functions (such as EJB definitions) and standard application interfaces. These definitions are emerging and maturing in several domains, including health care and manufacturing.

It is not clear whether the component movement will entirely replace the need to write some application code. To do so, it would be necessary to determine at the time the component is written all of the necessary interfaces, and all of the business logic, that would be needed by anyone who used the component. For some mature application areas and well-defined and accepted business practices it may well be possible to define the necessary interfaces and business logic. It is probably not possible for all application areas and interfaces. It is likely that businesses will customize some components to meet their particular needs.

What we have seen so far is that reusing application components and design patterns can greatly increase application developer productivity and reduce the time needed to bring an application to market. Not having to design, build, and test an entire application allows developers to meet the short time-to-market demands required in an e-business world.

SanFrancisco-based products are being delivered

Several companies are now delivering products based on IBM SanFrancisco. These range from applications to utilities and tools needed to build and execute applications. Front Ends, Inc., has used SanFrancisco to build a health-care industry application that handles billing, scheduling, and workflow. It is being used by a health-care network in Texas. They report that SanFrancisco provided a fivefold increase in programmer productivity in building the application.¹⁰ MindWork has created an end-to-end Web application that allows businesses to dynamically create Internet catalogs. This application has combined SanFrancisco components with the use of the IBM WebSphere* Application Server.¹¹ SKYVA International provides an application development environment that builds applications based on a company's business processes and business objects. It is using the tools to deliver a supply chain application for Sonopress, Inc., a Bertelsmann company, and a leading manufacturer of CDs.¹² InetSoft Technology Cor-

poration¹³ has updated its Style Report printing and report-generation solution to access data from SanFrancisco components.

These are just a few examples of recent activity. More details on some partners' projects are included in the papers. For the latest information on IBM SanFrancisco and its partners visit our Web site at: <http://www-4.ibm.com/software/ad/sanfrancisco/>.

Papers in this issue

The papers presented in this issue provide some insights on how the SanFrancisco team, its customers, and its partners are using the product to meet emerging market requirements.

VisualAge for Java continues to be an excellent development environment for building SanFrancisco applications. Additional tools are now also available for use in developing applications based on SanFrancisco. Two of these are discussed. "Business Component Prototyper for SanFrancisco," by Van Emde Boas-Lubsen, reviews a tool that can be used to rapidly build and test samples of components, and "Bridging the Framework Modeling and Implementation Gap," by Bunting, presents a partner's tool for developing SanFrancisco applications.

"The Use of IBM SanFrancisco Core Business Processes in Human Resources Scheduling," by Jauffmann and Logan, discusses the function reuse and development speed realized as a health care application was developed and deployed to support a real business. It also discusses how XML can be effectively used in SanFrancisco applications.

"Using JavaBeans Components as Accessors to Enterprise JavaBeans Components," by Tost and Johnson, discusses a client architecture that can be used with SanFrancisco applications today and allows an application to move to Enterprise JavaBeans in the future.

Finally, several customers continue to build applications on SanFrancisco. In "IBS Consist Two Years Later," by Van der Salm, we see how this company's use of SanFrancisco has evolved since the author's previous discussion.

Customers are beginning to make use of applications that are based on IBM SanFrancisco. At the same time, the product continues to change to respond to the needs of the market. These papers provide an updated picture of current activity.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc., or Object Management Group.

Cited references

1. K. A. Bohrer, "Architecture of the San Francisco Frameworks," *IBM Systems Journal* 37, No. 2, 156–169 (1998).
2. E. E. Inman, "Enterprise Modeling Advantages of San Francisco for General Ledger Systems," *IBM Systems Journal* 37, No. 2, 170–180 (1998).
3. R. A. Henders, "An Evolutionary Approach to Application Development with Object Technology," *IBM Systems Journal* 37, No. 2, 181–188 (1998).
4. R. L. van der Salm, "Introducing Shareable Frameworks into a Procedural Development Environment," *IBM Systems Journal* 37, No. 2, 200–214 (1998).
5. M. G. Polan, "Technical Note—Using the San Francisco Frameworks with VisualAge for Java," *IBM Systems Journal* 37, No. 2, 215–226 (1998).
6. R. Retallick and S. Sánchez, "Enterprise Knowledge Sharing, Activity Management, and a Fabric for Commitment," *IBM Systems Journal* 37, No. 2, 189–199 (1998).
7. See <http://www.w3.org/>.
8. See <http://www.w3.org/TR/NOTE-xlink-principles/>.
9. See <http://www.w3.org/TR/1999/NOTE-xml-schema-req-19990215/>.
10. See http://www-4.ibm.com/software/ad/sanfrancisco/v1_3frontends.html/.
11. See <http://www-4.ibm.com/software/ad/sanfrancisco/mindwork.html/>.
12. See http://www-4.ibm.com/software/ad/sanfrancisco/v1_3camelot.html/.
13. See <http://www.inetsoftcorp.com/index.html/>.

Accepted for publication December 28, 1999.

Verlyn M. Johnson *IBM Software Group, 3605 Highway 52 North, Rochester, Minnesota 55901 (electronic mail: verlyn@us.ibm.com)*. Dr. Johnson joined IBM in 1979. His background includes building and maintaining financial and manufacturing applications, designing and administering databases, and building application development tools. Currently he is working with the San Francisco group to help customers understand how to best use San Francisco, and to build relationships between the development team and customers to ensure that requirements are understood and met in future releases of the product.

David Kelly Austreng *IBM Software Group, 1200 5th Avenue, Seattle, Washington 98101 (electronic mail: dkstreng@us.ibm.com)*. Mr. Austreng joined IBM in 1998. He was recruited while presenting his new patent-pending Java technology for photographic 3-D presentation at the Internet World Exposition in Los Angeles. He has served as chairman and CEO of Blue Solutions, Computer Image Technology, and Net VR corporations. His background includes numerous inventions, including photographic presentation on the Macintosh™, which entered the market as "32-Bit QuickDraw." Today he is working on the e-components strategy for the IBM Software Group and helping to expand the possibilities of IBM San Francisco.